L2: Entry 8 of 16 File: USPT Jun 20, 2000

DOCUMENT-IDENTIFIER: US 6077312 A
TITLE: Apparatus, program product and method of debugging utilizing a context sensitive breakpoint

Detailed Description Text (29):
One suitable data structure for breakpoint table 62 is illustrated in greater detail in FIG. 4. Breakpoint table 62 includes a plurality of breakpoints 64 arranged in a linked-list data structure. Each breakpoint data structure 64 includes a key field 66, a good opcode field 68, a field 70 storing a pointer to a Dcode program, a next breakpoint field 72, and a Dcode program label field 74.

Detailed Description Text (32):
Breakpoint 64 also includes a pointer to a Dcode program, stored in field 70. The Dcode program, as discussed above, includes program code that is executed by Dcode interpreter module 60 for use in setting, updating, and/or processing a context sensitive breakpoint. Rather than a pointer to the program, the actual program code for the Dcode program may be stored within breakpoint 64. In addition, it should be appreciated that if data structure 62 supports unconditional breakpoints, the pointer may be set to NULL to indicate that no Dcode program is associated with the breakpoint. It should also be appreciated that, to handle other types of conditional breakpoints, pointer 70 may also point to dedicated Dcode programs for handling such conditional breakpoints.

Detailed Description Text (33):
Next breakpoint pointer field 72 is also provided in breakpoint 64 to provide a link to the next breakpoint 64 in data structure 62. It should be appreciated that the last breakpoint will include a NULL pointer in field 72. Also, field 72 may be omitted if breakpoints are stored in other data structures.

Detailed Description Text (79):
Next, in block 190, the Dcode program label is popped from the top of the stack, and in block 192, a breakpoint record (e.g., of the format of record 64 of FIG. 4), is added to the breakpoint table 62, including the address of the instruction to replace, the opcode currently stored at that address, a pointer to the Dcode program, and the program label that specifies the beginning of the test breakpoint routine in the Dcode program. In addition, in block 192, the Dcode program is saved in the Dcode interpreter module for future execution.

Detailed Description Text (80):
Next, once a record has been added to the breakpoint table, control passes to block 194 to replace the opcode in the executable program with a suitable breakpoint opcode that will trigger an exception during execution of the program, and consequently, handling of the breakpoint in the manner described herein.

Detailed Description Text (81):
Continuing with the above example, execution of the Dcode program of Table II results in a new breakpoint being added to the breakpoint table, with the breakpoint including an address corresponding to line 25 of the source code, a pointer to the Dcode program of Table II, the current opcode stored at line 25 of the source code, and the "TEST" label that specifies the start of the test breakpoint routine in the Dcode program.

Detailed Description Text (86):

interrupt that is handled by the breakpoint handler. As such, module 196 begins in block 200 by accessing the breakpoint table 62 to locate the breakpoint record 64 corresponding to the interrupting address. Once this record is located, block 202 determines whether a Dcode program exists, typically by accessing the pointer to the Dcode program at 70 (FIG. 4) and determining whether the pointer is set to NULL (indicating that no such program exists).

Previous Doc Next Doc Go to Doc#

```
5519879
5522076
5528572
5511205
5574893
5581708
5590357
5591950
5592677
5596764
5603017
5604488
5606714
5608854
5613149
5613143
5617082
5627888
5628217
5649096
5649208
5651099
5655146
5659734).pn.
(5659750
5664219
5666316
5671627
5689101
5703512
5719924
5732005
5737753
5754762
5764749
5764861
5764992
5774435
5790865
5793856
5797014
5815559
5821514
5835973
5835594
5838970
5847998
5852729
5856789
5856935
5859935
5867504
5872960
5871435
5875335
5881662
5898756
5900609
5901093
5907288
5911069
5919257
5926841
```

5930711
5937194
5938730
5940871
5943244
5958022
5969774
5974543
5978765
5982877
5991764).pn.